

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

JC542 U.S. PTO
09/209162
12/10/98


Page 1

```
1 on startMovie
2   global gLipsCursor
3
4   set gLipsCursor = 0
5   put " " into field "Message"
6   set the fontsize of member "Message" = 24
7
8   -- work around Director MIAW bug
9   tell the stage to emh_continue(#msgHandler)
10
11 end startMovie
12
13
14
15 on stopMovie
16   clearMessage()
17   set the member of sprite 2 = "lipsUp"
18   cursor 0
19 end
20
21
22 -- API Public Handlers -----
23 --- Ugly hack to work around problem with Director startup
24 --- of MIAWs. The problem is that, after calling a handler in the
25 --- MIAW, the StartMovie handler for the MIAW does not run until
26 --- the calling movie advances to its next frame.
27 --- Therefore, the calling sequence in the calling movie
28 --- has to be engineered so that the real handlers in the MIAW do not
29 --- run until after control has been transferred back to the calling
30 --- movie. However, at least one handler in the MIAW must be called
31 --- by the calling movie before the StartMovie handler will run.
32
33 --- startMeUp is the fake handler that, when called by the
34 --- main movie, will upon return to the main movie,
35 --- cause this movie's startMovie handler to run.
36
37 --- The second part of this wormy hack is contained in the MIAW's
38 --- startMovie handler... It is a call to a workAround handler in
39 --- the calling movie called continueComponent
40 --- The calling movie's continueRebus handler calls the real handlers
41 --- in the MIAW.
42
43 on emc_startMeUp
44   -- put "Macromedia sucks!"
45   return(1)
46 end startUp
47
48 -----
49
50 -----
51 --- initWindow is called by email main when a message handler
52 --- is opened
53 --
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 2

```
54
55 on emc_initWindow userName
56   global tm_RegisteredUsers, tm_UserName, tm_MailData
57
58   set tm_MailData = [:]
59   set tm_UserName = userName
60
61   tell the stage to emh_getRegisteredUsers()
62   set tm_RegisteredUsers = the result
63   fillToList(tm_RegisteredUsers)
64
65   -- put "EXIT initWindow" && "in frame" && the frame
66
67   return(1)
68 end initWindow
69
70 -----
71
72 --- closeWindow is not called unless Rebus plays as
73 --- a MIAW.
74
75 on emc_closeWindow
76   stopMovie
77   return(1)
78 end closeWindow
79
80 -----
81
82 on emc_getComponentInfo
83
84   -- the MIMEtype field will be application/txt
85
86   return( list( "Text", 2, #msgHandler, "text" ) )
87
88 end msh_getComponentInfo
89
90 -----
91
92
93 on msh_OpenMessage mailData, mode
94
95   return(openMessage(maildata, mode))
96
97 end msh_openMessage
98
99 -----
100
101
102
103 on msh_clearMessage
104
105   clearMessage()
106   return(1)
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 3

```
107 end msh_clearMessage
108
109 -----
110
111 on msh_sendMessage
112     global gMsgBody, tmG_mailData, gState, gMode
113
114     -- CLEAN RETURNS and QUOTES FROM TEXT MESSAGES (not needed in other
115     MIME types)
116
117     put "" into messageFinal
118     set messageVar = the text of field "Message"
119
120     repeat with i = 1 to the number of chars in messageVar
121
122         if char(i) of messageVar = RETURN then
123             put "$0D$" after messageFinal
124         else if char(i) of messageVar = QUOTE then
125             put "$2C$" after messageFinal
126         else
127             put char(i) of messageVar after messageFinal
128         end if
129
130     end repeat
131
132     -- PREPARE DATA STRUCTURES FOR EMAIL MAIN
133
134     setaProp tmG_mailData, #to, the text of field "To"
135     setaProp tmG_mailData, #from, the text of field "From"
136     setaProp tmG_mailData, #re, the text of field "Subject"
137     setaProp tmG_mailData, #status, #sent
138     setaProp tmG_mailData #date, the text of field "Date"
139     setaProp tmG_mailData, #msgbody, list(messageFinal)
140
141
142     -- SEND MESSAGE TO EMAIL MAIN
143     --- NOTE: sendToggle lets messageHandler know to send the message
144     --- rather than simply save a previously sent message
145
146     clearMessage()
147     -- alertSent()
148     return(tmG_mailData)
149
150 end msh_sendMessage
151
152 -----
153     --- This function called only when a message is already displayed
154     --- Therefore data structures should be already setup. We just
155     --- need to toggle the "to" and "from" fields.
156
157 on msh_replyMessage
158
159     global gMode, tmG_mailData
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 4

```
160      set gMode = #author
161      put getProp(tmG_mailData, #to) into field "From"
162      put getProp(tmG_mailData, #from) into field "To"
163
164      replyIt()
165
166      return(tmG_mailData)
167  end msh_replyMessage
168
169
170
171  on msh_PrintMessage
172
173      -- minimal implementation
174
175      printFrom the frame, the frame, 100
176
177      return(1)
178
179  end msh_PrintMessage
180
181
182
183  --- score script ss_goTheFrame
184
185  on exitFrame
186
187      go the frame
188
189  end
190
191
192  on flashSprite spriteNum
193      set the visible of sprite spriteNum = ~
194          not (the visible of sprite spritenum)
195
196      starttimer
197      repeat while the timer < 30
198          nothing
199      end repeat
200
201  end flashSprite
202
203
204
205  --- score scripts fr_sentmsg
206
207
208  on exitFrame
209      flashSprite(17)    --edit the "49" to be the sprite number
210          --for the sprite that has the upper left hand corner
211      "x"
212          --that is used to close the movie.
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 5

```
213     go to the frame
214 end
215
216
217 on mouseDown
218
219     -- make sure the flashing sprite is visible
220     set the visible of sprite 17 = TRUE
221     go to frame 2
222
223 end
224
225
226
227 --Utilities
228
229 -----
230     --- LocToWordPos returns the index of the word under loc in
231     --- the field. Vertical space in the field is assigned to the
232     --- closest char in the field. If loc is on a space between
233     --- words (horizontal only, see below), the function returns 0.
234     --- If the loc is not in the field, results are unpredictable.
235     --- The Lingo locToCharPos function will return either the first
236     --- or the last character in the field, depending upon the loc.
237     --- Comparable to the built-in Lingo locToCharPos function,
238     --- the location argument is assumed to be relative to the origin
239     --- of the text field.
240
241     --- Uses the Lingo function locToCharPos. locToCharPos is not
242     --- subject to the Lingo lineHeight bug. All locations with
243     --- vertical coordinates within a lineHeight return a character
244     --- on that line. For example, if lineHeight is 36 then any location
245     --- in the interval, [point(x,0) point(x, 36)] will return a character
246     --- on the first line, any location in the interval,
247     --- [point(x, 37) point(x, 72)] will return a character in the second
248     --- line, etc. Note: a character is returned even if the location
249     --- is on the white space between lines.
250
251 on locToWordPos fieldMember, locInField
252
253     set charPos = locToCharPos(fieldMember, locInField)
254
255     -- FIX THIS TO CHECK THAT LOC IS IN FIELD ELSE RETURN 0.
256     -- locTocharPos will return 1 if the loc is above or left of the field
257     -- It will return lenght(text) + 1 if the loc is below or right of
258     field
259
260     if char charPos of field fieldMember = " " then -- on a space
261         return(0)
262     else
263         return(the number of words in char 1 to charPos of field
264         fieldMember)
265     end if
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 6

```
266
267 end locToWordPos
268
269 -----
270
271 on fillToList userNames
272
273 put "" into field "ToDown"
274 repeat with uname in userNames
275   put uname & RETURN after field "ToDown"
276 end repeat
277
278 end fillToList
279
280
281 -----
282
283 on alertSent
284   go to frame "fr_sentMsg"
285 end alertsent
286
287
288 on saveIt
289
290   global tmG_messageNumber, tmG_mailData, gMode
291
292
293   -- Put fields into variables
294
295   put field "To" into toVar
296   put field "Subject" into subjectVar
297   put field "Message" into messageVar
298
299
300   -- CLEAN RETURNS FROM TEXT MESSAGES (not needed in other MIME types)
301
302   put "" into messageFinal
303
304   repeat with i = 1 to the number of chars in messageVar
305
306     if char(i) of messageVar = RETURN then
307       put "$0D$" after messageFinal
308     else if char(i) of messageVar = QUOTE then
309       put "$2C$" after messageFinal
310     else
311       put char(i) of messageVar after messageFinal
312     end if
313
314   end repeat
315
316
317   -- PREPARE DATA STRUCTURES FOR EMAIL MAIN
318
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 7

```
319      setaProp tmG_mailData, #to, toVar
320      setaProp tmG_mailData, #re, subjectVar
321      setaProp tmG_mailData, #msgbody, messageFinal
322
323
324      -- SEND MESSAGE TO EMAIL MAIN
325
326      set sendToggle = 0
327
328      tell the stage
329
330          global tmG_messageNumber, tmG_mailData
331
332          messageHandler sendToggle
333
334      end tell
335
336
337  end
338
339  on replyIt
340
341      global tmG_messageNumber, tmG_mailData, gMode
342
343      -- Swap the "to" and "from" fields
344      -- Put header information into variables and clear message number
345
346      put the abbreviated date into field "Date"
347      put "Re: " & getaProp(tmG_mailData,#re) into field "Subject"
348      set the editable of member "Message" to TRUE
349      put"---- " & getProp(tmG_mailData, #to) & "'s Note ----" & RETURN -
350          before field "Message"
351      put RETURN before field "Message"
352
353  end
354
355
356
357  -- script of cast member closeWindow
358
359  on mouseUp
360
361      -- Should check whether user wants to Save the message
362      --- saveIt
363
364      tell the stage to emh_killComponent()
365
366  end
367
368
369
370
371
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 8

```
372 --- script of cast member read Message Button
373
374 on mouseUp
375   speak(the text of member "Message")
376 end
377
378
379
380 on openMessage mailData, mode
381
382   global gToBoxDown, tm_RegisteredUsers
383   global tmG_mailData, gMode
384
385   set tmG_mailData = maildata
386   set gMode = mode
387
388   -- Pull out the info from tmG_mailData and place where needed
389
390   put getaProp(tmG_mailData,#to) into field "To"
391   put getaProp(tmG_mailData,#from) into field "From"
392   put getaProp(tmG_mailData,#re) into field "Subject"
393   put getaProp(tmG_mailData,#date) into field "Date"
394
395   put getaProp(tmG_mailData,#status) into statusVar
396
397   set msgBody = getaProp(tmG_mailData,#msgbody)
398   if count(msgBody) = 0 then
399     put "" into messageVar
400   else
401     put getAt(msgbody, 1) into messageVar
402   end if
403
404   -- Reinstate returns and quotes into the message
405
406   put "" into messageFinal
407
408   repeat with i = 1 to the number of words in messageVar
409
410     if word(i) of messageVar = "$0D$" then
411       put RETURN after messageFinal
412     else if word(i) of messageVar = "$2C$" then
413       put QUOTE after messageFinal
414     else
415       put " " & word(i) of messageVar & " " after messageFinal
416     end if
417
418   end repeat
419
420   put messageFinal into field "Message"
421
422
423   -- allow or disallow user changes (gToBoxDown is for the "To" field)
424
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 9

```
425 if gMode = #display OR statusVar = #sent then
426   set the editable of member "Message" to FALSE
427   set the editable of member "Subject" = FALSE
428   set gToBoxDown = 0
429 else if gMode = #author then
430   -- else if statusVar = #saved then
431   fillToList(tm_RegisteredUsers)
432   set the editable of member "Message" to TRUE
433   set the editable of member "Subject" = TRUE
434   set gToBoxDown = 1
435 end if
436
437   return(1) -- need error checking and return meaningful val
438 end
439
440
441 on clearMessage
442
443   global tmG_mailData, gToBoxDown
444   -- CLEAR FIELDS
445
446   put "" into field "To"
447   put "" into field "Date"
448   put "" into field "Subject"
449   put "" into field "Message"
450
451   -- NEXT TWO FIELDS NOT CLEARED SO THAT USER CAN SEND
452   -- MULTIPLE MESSAGES WITHOUT CLOSING THE MOVIE
453   -- put "" into field "ToDown"
454   -- put "" into field "From"
455
456   -- NOT SURE ABOUT WHAT TO DO ABOUT MAILDATA?????
457
458   -- SET UP TO BEGIN A NEW MESSAGE
459
460   set the editable of member "Message" = TRUE
461   set the editable of member "Subject" = TRUE
462   set gToBoxDown = 1
463   cursor 0
464
465 end clearMessage
466
467
468
469 -- script of cast member To
470
471 on mouseUp
472
473   global gToBoxDown, gLipsCursor
474
475   -- Pull down student field: change field from
476   -- up (member 11) to down (member 12)
477
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 10

```
478
479     if gLipsCursor then
480         speak(the text of field "To")
481
482     else if gToBoxDown = 1 then
483         set the member of sprite 11 to member "ToDown"
484     end if
485
486 end
487
488
489 -- script of cast member toDown
490
491 on mouseUp
492
493     -- KEEP TRACK OF SELECTED NAME
494
495     set whoTo = word 1 of line(the mouseLine) of field "ToDown"
496
497     -- Put selected user name into up version of student field and
498     switch the
499     -- field from down (member 12) to up (member 11)
500
501     put whoTo into field "To"
502
503     set the member of sprite 11 to member "To"
504
505 end
506
507
508 -- script of cast member From
509
510 on mouseUp
511     global gLipsCursor
512
513     if gLipsCursor then
514         speak(the text of field "From")
515     end if
516 end
517
518
519
520 -- script of cast member Date
521
522 on mouseDown
523     global gMode, gLipsCursor
524
525     if field "Date" = EMPTY and gMode = #author then
526         put the abbreviated date into field "Date"
527     end if
528
529     if gLipsCursor then
530         speak(the text of field "Date")
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 11

```
531     end if
532 end
533
534
535
536 -- script of cast member Subject
537
538 on mouseUp
539   global gLipsCursor
540
541   if gLipsCursor then
542     speak(the text of field "Subject")
543   end if
544
545 end
546
547
548 -- script of cast member Message
549
550 on mouseDown
551   global gLipsCursor
552
553   if gLipsCursor then
554     set textOrigin = the loc of sprite 15
555     set upLoc = point(the mouseH, the mouseV) - textOrigin
556     set myword = loctowordpos(member "Message", upLoc)
557
558     if myword > 0 then
559       set astr = word myword of field "Message"
560       speak( astr)
561     else alert "Click on a word." & RETURN & "I will read it."
562   end if
563
564 end
565
566
567
568 --- script of cast member lipsDown
569
570 on mouseUp
571   global glipsCursor, gmode
572
573   -- reset the lipsCursor
574   cursor 0
575   if gmode = #author then
576     -- set the editable of member "To" = TRUE
577     set the editable of member "Subject" = TRUE
578     set the editable of member "Message" = TRUE
579   end if
580   set gLipsCursor = 0
581   set the member of sprite 2 = "lipsUp"
582
583 end
```

**Appendix C: KidCode® Lingo Client/Server Text Message
Handling Component Scripts**

Page 12

```
584
585
586 --- script of cast member lipsUp
587
588 on mouseUp
589   global glipsCursor
590
591 -- reset the lipsCursor
592 cursor [27]
593
594 set the editable of member "To" = FALSE
595 set the editable of member "Subject" = FALSE
596 set the editable of member "Message" = FALSE
597
598 set gLipsCursor = 1
599 set the member of sprite 2 = "lipsDown"
600
601 end
```